ESIP Technology Evaluation Framework Final Recommendations

John Graybeal
Graybeal.SKI Consulting

Summary of Product

The delivered product provides a Google spreadsheet template addressing Software evaluation at 9 NASA Technology Readiness Levels. Spreadsheet features include category summaries; binary, normalized (0.0 to 1.0), or positive value (>0.0) ratings; TRL-based line item weightings; page and category averages; line item and page-wide comments; and instant re-calculation for different TRL levels. The template includes source references and some user documentation.

Recommendations: Evaluation Spreadsheet (General)

Product Recommendations. The Google spreadsheet format worked very well for all the requested features, including individual item and summary information and comments.

When looking at a spreadsheet instance, or different copies of the template, the actual document is totally unidentifiable from the content. Add a first page to include instance metadata (optionally present summary data from the evaluation, like how much was completed, and resulting ratings).

User documentation is hard to present well in this form, but more could be in the form itself (rather than in references). The references to source material could be in every spreadsheet instance; at a minimum they should be referenced explicitly on the documentation page.

Create and reference developer documentation elsewhere (in-line is hard, and not critical).

Some minor usability issues relate to customizing the TRL: (a) changing the target TRL should automatically update the display lines (there is no way to implement that), and (b) locking the TRL cell, or creating a TRL-specific spreadsheet from the template, could be valuable. Another possible issue is the difficulty of gracefully locking the non-data cells, to prevent user modification of the algorithm.

Process Recommendations. The spreadsheet template offers great opportunity for customization before it is distributed to the evaluators; the client's evaluation manager should be responsible for understanding the spreadsheet and potential changes. (The recommended technique is by changing the weightings, for easiest comparison with the template.) This is a very important process, which has not yet been tried or evaluated.

Many evaluators identified lack of expertise, or lack of access to information, as an obstacle to filling out many questions. One option to partly address this is to allow the project team to pre-fill, or provide information about, highly technical or unpublished data. The evaluators could modify the answers based on the information the project provided.

More active hand-holding of evaluators by ESIP support personnel could improve understanding of goals, questions, and optimal procedures. Many of the questions and categories would benefit from additional explanatory text; conceivably the main text should be shorter, and longerexplanations available as help (on request).

Keep track of all customized and completed instances. These can be analyzed later to learn which criteria are most often deleted during customization, not filled out during evaluation, not commented upon, or not performed. These are useful hints as to the least valuable questions, and/or the hardest goals to achieve. Perhaps more significantly, the customizations that **add** evaluation criteria can be used to create additional best practices or evaluation types.

Maintenance Recommendations: In a similar vein, processes for version control — tracking changes over time, and during an evaluation process, and documenting both of these for ready access — need further consideration to avoid losing change control of the software.

In the long term, it's likely this capability will be more effectively provided as an on-line system, but much more experience is needed before that happens. Keeping all the evaluation types in one template is critical for maintainability in this form (because making the same core changes identically across multiple Google spreadsheets will be unmaintainable). Therefore, a critical transition will occur when the number of custom evaluation types can't easily be supported by a single Google spreadsheet template; try to delay this transition as long as possible.

Recommendations: Software Evaluation

Timeliness: Criteria should be updated (every 2 years?) to reflect current practices.

Checklist length: A decision must be reached about checklist length. Possibly two checklists will be appropriate: a 'long-form' checklist, to provide a complete capture of best practices, and a 'short-term' version, consolidating multiple related questions into single items. I suspect that the long-form version will provide a more accurate assessment, and would be appropriate for more systematic evaluations. The long-term checklist is also important because it guides product managers and developers more directly to better practices.

Validation with other criteria: Sustainable Software Software Evaluation recommendations from 2011 were key to efficient creation of first spreadsheet. Evaluation of the resulting criteria against other standards (ESTO, Software Sustainability Institute, previous TRL checklists, software quality criteria, etc.) could yield validation and further improvements.

Software classes: 3 classes of software are *applications*; *web services*; and *infrastructures*. Some customization of the Software TEF for each is possible; but the customization for a particular instance is probably equally detailed, and more directly useful. On balance, more concrete analysis/justification is needed before increasing the complexity by adding class-based customizations.

Recommendations: Other Evaluation Types

The work done so far in the product for non-software evaluation types simply identifies which software criteria also apply to the evaluation types listed below

Data Sets: The evaluation criteria for data sets — and the definition of best practices in creating them — are less universally agreed, and more dependent on the purpose, format, and distribution technique of the data. As data set criteria are developed further, they will inevitably be strongly aligned with metadata best practices, which have been extensively discussed and documented. Data set best practices beyond the realm of metadata will be more varied and difficult to frame for the Technology Evaluation Framework.

References to be checked in this area include Bruce/Hillman 2004, W3C's Data on the Web Best Practices Working Group and Data Quality Vocabulary, RDA's and ESIP's many data-related Working Groups, Digital Curation Committee's Metadata Standards and RDA's Community Metadata Directory. The obvious challenge is to filter all this advice and information into best practices that can be easily evaluated using a TEF spreadsheet; I recommend the approach of trying to find those collections of high-level criteria that are already in a usable form, and blend those for evaluation in actual use with the spreadsheet.

Knowledge Artifacts: The evaluation criteria for knowledge artifacts — and the definition of best practices in creating them — are even less universally agreed, and more dependent on the nature of the artifact, than those for data sets. ESIP's Semantic Committee may be in the best positions to identify and integrate the various viewpoints on the best practices for knowledge artifacts. As with data sets, reviewing the various viewpoints and analyses — several ontology evaluation criteria papers have been published recently — will take some time.

The best approach for the TRL Evaluation process to follow to bring this information into the Technology Evaluation Framework may be to wait for the request to evaluate specific knowledge artifacts (e.g., vocabularies or ontologies), and customize/add evaluation criteria expressly for that type of artifact.

Conclusion and Potential

The existing Technology Evaluation Framework has been useful in evaluating the technology readiness of software projects, and has a strong potential to provide project managers with useful indications of what their software projects should include as they mature.

In fact, the number of potential applications is quite large, including:

- providing a basis to discuss project evaluation or deliverable requirements with clients,
- creating project readiness/progress evaluation criteria for software managers,
- tracking a project's technology readiness levels throughout its life cycle,
- providing a long-term database of project results, leading to potential improvements in development, management, and evaluation practices (including to TEF itself),
- serving as a foundation of technology maturity evaluation service offerings, and
- providing an up-to-date summary of best practices for project engineering practitioners.

If the use of the Technology Evaluation Framework grows, it should be evaluated against its own criteria to identify improvements that will be needed for a more robust evaluation framework.

For a first prototype of a sophisticated tool, the Technology Evaluation Framework has proven an extremely valuable project and product, with many lessons already learned and many opportunities for enhancement and reuse.

Appendix: Ideas for Improvements

In addition to the above comments, the following improvement ideas were identified during the project.

- Generate a nicely formatted summary report for each section,
- Aggregate summary reports across all the evaluations of a project.
- Let each reviewing team member independently fill out a separate form, then provide a report on the merged team results (taking average or mode of ratings, merging comments, and highlighting disagreements). Teams could then review the merged results together.
- Automatically provide ‘retrograde’ analysis, where the answers are used to evaluate against lower or higher TRL levels (so 42% at TRL 5, could be 55% for TRL 4, 73% for TRL 3, etc.). This could suggest the actual TRL level of the product.
- Provide a progress bar showing completion progress for the form. (This would be appropriate on the metadata page mentioned in the General section.)
- The opportunity for feedback on each question could be emphasized more strongly in the form’s layout.
- If the form includes a place for the project’s estimated TRL level (either filled out when the form is filled out, or later based on the higher-level evaluation), it could provide valuable contextual information for the database of evaluation instances.